~ ~ Author/ Inventor Search - Patents

^ 2/5/2 (Item 2 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0014593791 *Drawing available*
WPI Acc no: 2004-775756/200476
XRPX Acc No: N2004-611110
File processing method involves regenerating header files controlled using data selected preprocessed file, so that data in selected file controls addition of data in regenerated header file
Patent Assignee: KONINK PHILIPS ELECTRONICS NV (PHIG)
Inventor: **POPP O**; ONDREJ P

Patent Family (7 patents, 107 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
WO 2004092953	A2	20041028	WO 2004IB50425	A	20040413	200476	B
EP 1627300	A2	20060222	EP 2004727084	A	20040413	200615	E
			WO 2004IB50425	A	20040413		
KR 2006009843	A	20060201	WO 2004IB50425	A	20040413	200660	E
			KR 2005719534	A	20051014		
JP 2006523887	W	20061019	WO 2004IB50425	A	20040413	200669	E
			JP 2006506834	A	20040413		
CN 1860437	A	20061108	CN 200480010025	A	20040413	200720	E
US 20070067316	A1	20070322	WO 2004IB50425	A	20040413	200723	E
			US 2005552777	A	20051012		
CN 100437476	C	20081126	CN 200480010025	A	20040413	200936	E

Priority Applications (no., kind, date): EP 2003101032 A 20030416

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
WO 2004092953	A2	EN	27	5		
National Designated States,Original	AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NA NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW					
Regional Designated States,Original	AT BE BG BW CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IT KE LS LU MC MW MZ NL OA PL PT RO SD SE SI SK SL SZ TR TZ UG ZM ZW					
EP 1627300	A2	EN			PCT Application	WO 2004IB50425
					Based on OPI patent	WO 2004092953
Regional Designated	AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IT LI LT LU LV MC MK NL PL PT RO SE SI SK TR					

States,Original						
KR 2006009843	A	KO			PCT Application	WO 2004IB50425
					Based on OPI patent	WO 2004092953
JP 2006523887	W	JA	22		PCT Application	WO 2004IB50425
					Based on OPI patent	WO 2004092953
US 20070067316	A1	EN			PCT Application	WO 2004IB50425

Alerting Abstract WO A2
NOVELTY - The source files (20a-20c) are expanded using data from a header file (21) for generating respective preprocessed files (23a-23c). A preprocessed file is selected based on detection of selected file elaborating data from header file. The headers (26a-26c) controlled using data from selected file, is regenerated so that information in selected file controls addition of data in regenerated header files.
DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

1. computer program product for processing file; and
2. file processing program.

USE - For processing files in computer.
ADVANTAGE - Enables processing the files efficiently and separately, after generation of the source and header files.
DESCRIPTION OF DRAWINGS - The figure shows a schematic view of the file processing system.
20a-20c original source files
21 original header file
23a-23c preprocessed files
24a collective processing stage
25a-25c source files
26a-26c header files
Title Terms /Index Terms/Additional Words: FILE; PROCESS; METHOD; REGENERATE; HEADER; CONTROL; DATA; SELECT; SO; ADD

2/5/5 (Item 1 from file: 348)
DIALOG(R)File 348: EUROPEAN PATENTS

01830922

REGENERATING HEADER FILES OUT OF PREPROCESSED AND AFTERWARDS MODIFIED SOURCE FILES
REGENERIERUNG VON HEADER DATEIEN AUS VORVERARBEITETEN UND ANSCHLIESSEND MODIFIZIERTEN SOURCE DATEIEN
TRAITEMENT D'UN PROGRAMME INFORMATIQUE COMPILABLE

Patent Assignee:

- **Koninklijke Philips Electronics N.V.** (200769)
  Groenewoudseweg 1; 5621 BA Eindhoven (NL)
  (Applicant designated States: all)

Inventor:

- **POPP, Ondrej**
  c/o Prof. Holstlaan 6; NL-5656 AA Eindhoven; (NL)

Legal Representative:

- **Eleveld, Koop Jan (135781)**
  Philips Intellectual Property & Standards, P.O. Box 220; 5600 AE Eindhoven; (NL)

	Country	Number	Kind	Date	
Patent	EP	1627300	A2	20060222	(Basic)
	WO	2004092953		20041028	
Application	EP	2004727084		20040413	
	WO	2004IB50425		20040413	
Priorities	EP	2003101032		20030416	

? 2/5/6 (Item 1 from file: 349)
DIALOG(R)File 349: PCT FULLTEXT

01169933

REGENERATING HEADER FILES OUT OF PREPROCESSED AND AFTERWARDS MODIFIED SOURCE FILES
TRAITEMENT D'UN PROGRAMME INFORMATIQUE COMPILABLE

Patent Applicant/ Patent Assignee:

- **KONINKLIJKE PHILIPS ELECTRONICS NV**
  Groenewoudseweg 1, NL-5621 BA Eindhoven; NL; NL (Residence); NL (Nationality); (For all designated states except: US)

Patent Applicant/ Inventor:

- **POPP Ondrej**
  c/o Prof. Holstlaan 6, NL-5656 AA Eindhoven; NL; NL (Residence); NL (Nationality); (Designated only for: US)

Legal Representative:

- **ELEVELD Koop J (agent)**
  Prof. Holstlaan 6, NL-5656 AA Eindhoven; NL

	Country	Number	Kind	Date
Patent	WO	200492953	A2-A3	20041028
Application	WO	2004IB50425		20040413
Priorities	EP	20031010325		20030416

English Abstract:

A plurality of source files (20) and one or more header files (21) are provided. The header files that contain information that several of the source files refer to. The original files are preprocessed, each to generate a respective preprocessed file, said preprocessing comprising expanding the several of the source files with the information from a first header file. A collective processing step (24a) is applied to make coordinated changes to information from the preprocessed files. The changed preprocessed files are used to regenerate modified source (25) and header (26) files. For regenerating (24b) the first header file, one of the preprocessed files is selected on the basis of detection that the selected file elaborates the information from the first header file. The first header file (26a) is regenerated from the information in the selected file.

10552777 Processing of a Compileable Computer Program - Results

~ ~ Author/ Inventor Search - NPL

^ 6/3,K/1 (Item 1 from file: 2)
DIALOG(R)File 2: INSPEC

04821099
Title: A general-purpose programmable video signal processor
Author(s): van Roermund, A.H.M.; Hemeryck, C.G.; Huizer, C.M.; **Popp, O.**; Schmitz, J.M.P.; Snijder, P.J.
Author Affiliation: Philips Res. Labs., Eindhoven, Netherlands
Inclusive Page Numbers: 339-46
Publisher: Elsevier, Amsterdam
Country of Publication: Netherlands
Publication Date: 1990
Conference Title: Signal Processing of HDTV, II. Proceedings of the Third International Workshop on HDTV
Conference Date: 30 Aug.-1 Sept. 1989
Conference Location: Turin, Italy
Editor(s): Chiariglione, L.
ISBN: 0 444 88833 0
Number of Pages: xx+899
Language: English
Subfile(s): B (Electrical & Electronic Engineering); C (Computing & Control Engineering)
INSPEC Update Issue: 1991-005
Copyright: 1991, IEE
Author(s): van Roermund, A.H.M.; Hemeryck, C.G.; Huizer, C.M.; **Popp, O.**; Schmitz, J.M.P.; Snijder, P.J.
Descriptors: **computerised** picture processing; video signals

~ ~ Patent Literature Abstracts

8/3,K/2 (Item 2 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0018875422 *Drawing available*
WPI Acc no: 2009-F55003/200925
Method for extracting compile object file, by producing header file that corresponds to change item, and producing header file by using extracted source file
Patent Assignee: KTF TECHNOLOGIES INC (KTFT-N)
Inventor: KIM D K

Patent Family (1 patents, 1 countries)							
Patent Number	**Kind**	**Date**	**Application Number**	**Kind**	**Date**	**Update**	**Type**
KR 2009011974	A	20090202	KR 200776065	A	20070727	200925	B

Priority Applications (no., kind, date): KR 200776065 A 20070727

Patent Details					
Patent Number	**Kind**	**Lan**	**Pgs**	**Draw**	**Filing Notes**
KR 2009011974	A	KO	13	5	

Alerting Abstract ...NOVELTY - The method involves: producing a source file list in which the **header file** is defined; extracting a **source file including** the code for changing the **header file** from the source file list; extracting a change item; and producing the **header file** that corresponds to the extracted change item; and producing the **header file** by using the extracted source file.

Original Abstracts:object file sampling method. The compile object file sampling method according to the invention is characterized by comprising the source file searching stage: extracting the **source file including** the code which the **header file** is changed; it extracts the change item of the source file list **generating** step: **header file** producing the source file list in which the **header file** is announced; and corresponds to the change item from the source file list and the compile target file list **generating** step which produces the **header file** much compile target file list by using the extracted source file as described above. Therefore, the invention has the advantage that it prevents in beforehand that the excessive compilation time according to the compile of the unnecessary source file is required.The compile, preprocessing, source **file**, **header file**, extraction.Image 1/1

Claims:[CLAIM 1] The compile object file sampling method including the compile target file list **generating** step which produces the **header file** much compile target file list by using the source file searching stage extracting the source file from the source file list, and the extracted source file as described above of the method for extracting the compile object file comprising the code which the **header file** is changed; it extracts the change item of the source file list **generating** step: **header file** producing the source file list in which the **header file** is announced; and corresponds to the change item... ...CLAIM 2] The compile object file sampling method of claim 1, wherein in the source file list **generating** step, the change yes or no of the **header file** is determined according to the file size of the corresponding **header file** and change yes or no of the file version... ...CLAIM 3] The compile object file sampling method of claim 1, wherein the change item is extracted with the recent Version of the corresponding **header file** through immediately, the difference comparison on code between the former version; and the change item is comprised of the

changed variable, and the character string... ...including the compile object file duplication decimation stage of claim 1, wherein the predetermined duplication pointer is set up after the compile target file list **generating** step so that the duplication compile be performed about the specific source file...

8/3,K/3 (Item 3 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0018377173 *Drawing available*
WPI Acc no: 2008-M97509/200876
Header file preprocessing method for c-language, involves accessing header file based on header file path information, and including header file in source file, if header file is found in header file table
Patent Assignee: DAEGU GYEONGBUK INST SCI & TECHNOLOGY (DAEG-N)
Inventor: LEE D H

Patent Family (1 patents, 1 countries)							
Patent Number	**Kind**	**Date**	**Application Number**	**Kind**	**Date**	**Update**	**Type**
KR 2008057422	A	20080625	KR 2006130689	A	20061220	200876	B

Priority Applications (no., kind, date): KR 2006130689 A 20061220

Patent Details					
Patent Number	**Kind**	**Lan**	**Pgs**	**Draw**	**Filing Notes**
KR 2008057422	A	KO	10	3	

Alerting Abstract ...NOVELTY - The method involves reading a **header file** table from a file, if the file storing the table is found, or **generating** the table, if the file is not found. The table is stored in the file, if a target **header file** is not found. The table is checked for the **header file**, if the **header file** is found. The target **header file** is accessed based on **header file** path **information** and is **included** in a **source file**, if the **header file** is found in the table. The target **header file** is searched, and the path **information** is stored and **included** in the **source file**, if the target **header file** is not found in the table. USE - Method for preprocessing a **header file** of c-language... ...ADVANTAGE - The method stores **header file** path **information** included in the source or **header file** in the **header file** table, stores the **header file** table as the file, and avoids repeating the search about the **header file** in different source files, thus improving the compilation performance, increasing the simplicity and reducing the time for searching the **header file**. ...

Original Abstracts: The present invention is to provide the method of pre-treating which is the invention about the **header file** method of pre-treating for the compile performance improvement, and one side of the present invention is performed among the complie process in the source file as to the method of pre-treating for including the **header file** in case of being determined as thing, and the file in which (a) **header file** table is stored does not exist the invention produces the file in which the **header file** table searching the route of the **header file** and includes the routing **information** of the **header file** is stored, and the invention is performed in case of being determined as thing, and the file in which the step that includes and (b) **header file** table are stored in the source file exists the **header file** accesses the **header file** according to the routing **information** of the stored **header file** table, and includes the step including the **header file** in the source file.Image 1/1

Claims:[CLAIM 1] As to the method of pre-treating for including the **header file** in the source file among the complie process, it determines whether the file in which (a) **header file** table is stored exists. In case of being determined as thing, exists the **header file** table is read on file. It determines whether file and the step: **header file** to be **included** in (b) **source file** exist produces the **header file** table in case of being determined as thing. Does not exist in case of being determined as thing, does not exist the **header file** table is stored to file. In case of being determined in the step: (c) (b) step that determines whether the **header file** to be included as described above exists in the **header file** table in case of being determined as thing or not exists as thing, the **header file** to be included as described above exists in the **header file** table it is performed. And it accesses according to the **header file** routing **information** of the **header file** table in the **header file** to be included as described above. In case of being determined in the step including the **header file** to be included as described above in the source file and (d) (b) step as thing, the **header file** to be included as described above does not exist in the **header file** table it is performed. The method of pre-treating searching the **header file** to be included as described above, and storing the **header file** routing **information** of the **header file** to be included as described above in the **header file** table, and for including the step including the **header file** to be included as described above in the source file... ...is determined as thing; and for more including (b) step or the step repeating (d) step of claim 1, wherein in (e) (b) step, the **header file** to be **included** in the **source file** more does not exist... ...CLAIM 3] The method of pre-treating for including the step included in step, the step that is performed in case the final modification **information** of (c2) **header file** table and real final modification **information** of the **header file** does not coincide with; and renews the final modification **information** of the **header file** table and inclusion **header file information** of the **header file** table, and the source **file** the **header file** to (c3) above statement has to be included of claim 1 or 2, wherein (c) step accesses in the **header file** to be included as described above according to the **header file** routing **information** of (c1) **header file** table... ...CLAIM 4] The method of pre-treating for including the step included in step, the **header file** routing **information** of the **header file** to be included as described above in (d2) **header file** table, and the final modification **information** and the step that the inclusion **header file information** is stored, and the source **file** the **header file** to (d3) above statement has to be included of claim 3, wherein (d) step searches the **header file** to (d1) above statement has to be included...

^ 8/3,K/11 (Item 11 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0015084825 *Drawing available*
WPI Acc no: 2005-434281/200544
Related WPI Acc No: 2005-434259; 2005-541252; 2007-794883
XRPX Acc No: N2005-352458
Application program schema language transformation method involves transforming schema language into two different files
Patent Assignee: MICROSOFT CORP (MICT)
Inventor: FLEEGAL E B

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 20050125811	A1	20050609	US 200276667	A	20020214	200544	B
			US 200538554	A	20050119		

Priority Applications (no., kind, date): US 200276667 A 20020214; US 200538554 A 20050119

Patent Details						
Patent Number	**Kind**	**Lan**	**Pgs**	**Draw**	**Filing Notes**	
US 20050125811	A1	EN	48	26	Division of application	US 200276667

Alerting Abstract ...of application program interface (API) written in API schema language. The schema language is transformed into two different files such as component object model (COM) **header file**, mapping file, proxy object code file and documentation file. ...DESCRIPTION OF DRAWINGS - The figure shows the flowchart illustrating exemplary process for **generating** output files based on application program interface schema language (APISL) document.

...**Original Abstracts:**techniques allow an application programming interface to be defined in a markup language. This interface definition can then be transformed into a non-markup language **source file**. Examples of transformation techniques **include** techniques to transform the interface definition into a component object model (COM) **header file**, a documentation file that describes the contents of the non-markup language source file, and a proxy object code file.

Claims:1. A method comprising:receiving an indication of an interface definition, wherein the interface definition includes a plurality of constructs;transforming the interface definition into **data** for a first file; andtransforming the interface definition into **data** for a second file, wherein the **data** for the first file is different than the **data** for the second file.

8/3,K/12 (Item 12 from file: 350) (Note current app)

0014593791 *Drawing available*
WPI Acc no: 2004-775756/200476
XRPX Acc No: N2004-611110
File processing method involves regenerating header files controlled using data selected preprocessed file, so that data in selected file controls addition of data in regenerated header file
Patent Assignee: KONINK PHILIPS ELECTRONICS NV (PHIG)
Inventor: POPP O; ONDREJ P

Patent Family (7 patents, 107 countries)							
Patent Number	**Kind**	**Date**	**Application Number**	**Kind**	**Date**	**Update**	**Type**
WO 2004092953	A2	20041028	WO 2004IB50425	A	20040413	200476	B
EP 1627300	A2	20060222	EP 2004727084	A	20040413	200615	E
			WO 2004IB50425	A	20040413		
KR 2006009843	A	20060201	WO 2004IB50425	A	20040413	200660	E
			KR 2005719534	A	20051014		
JP 2006523887	W	20061019	WO 2004IB50425	A	20040413	200669	E
			JP 2006506834	A	20040413		
CN 1860437	A	20061108	CN 200480010025	A	20040413	200720	E

US **20070067316**	A1	20070322	WO 2004IB50425	A	20040413	200723	E
			US 2005552777	A	20051012		
CN 100437476	C	20081126	CN 200480010025	A	20040413	200936	E

Priority Applications (no., kind, date): EP 2003101032 A 20030416

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
WO 2004092953	A2	EN	27	5	
National Designated States,Original	AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NA NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW				
Regional Designated States,Original	AT BE BG BW CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IT KE LS LU MC MW MZ NL OA PL PT RO SD SE SI SK SL SZ TR TZ UG ZM ZW				
EP 1627300	A2	EN			PCT Application: WO 2004IB50425
					Based on OPI patent: WO 2004092953
Regional Designated States,Original	AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IT LI LT LU LV MC MK NL PL PT RO SE SI SK TR				
KR 2006009843	A	KO			PCT Application: WO 2004IB50425
					Based on OPI patent: WO 2004092953
JP 2006523887	W	JA	22		PCT Application: WO 2004IB50425
					Based on OPI patent: WO 2004092953
US 20070067316	A1	EN			PCT Application: WO 2004IB50425

Alerting Abstract ...NOVELTY - The **source files** (20a-20c) are **expanded** using **data** from a **header file** (21) for **generating** respective preprocessed files (23a-23c). A preprocessed file is selected based on detection of selected file elaborating **data** from **header file**. The **headers** (26a-26c) controlled using **data** from selected file, is **regenerated** so that **information** in selected file controls addition of **data** in **regenerated header files**. ... ADVANTAGE - Enables processing the files efficiently and **separately**, after **generation** of the source and **header files**.... ... 21 original **header file** ... ... 26a-26c **header files Title Terms** .../Index Terms/Additional Words: **REGENERATE**; **HEADER**; ... ...**DATA**; **Class Codes**

Original Abstracts:A plurality of source files and one or more **header files** are provided. The **header files** that contain **information** that several of the source files refer to. The original files are preprocessed, each to **generate** a respective preprocessed file, said preprocessing comprising **expanding** the several of the **source files** with the **information** from a first **header** 5 **file**. A collective processing step is applied to make coordinated changes to **information** from the preprocessed files. The changed preprocessed files are used to **regenerate** modified source and **header files**. For **regenerating** the first **header file**, one of the preprocessed files is selected on the basis of detection that the selected file elaborates the **information** from the first **header file**. The first **header file** is

regenerated from the **information** in the selected file... ... A plurality of source files and one or more **header files** are provided. The **header files** that contain **information** that several of the source files refer to. The original files are preprocessed, each to **generate** a respective preprocessed file, said preprocessing comprising **expanding** the several of the **source files** with the **information** from a first **header 5 file**. A collective processing step is applied to make coordinated changes to **information** from the preprocessed files. The changed preprocessed files are used to **regenerate** modified source and **header files**. For **regenerating** the first **header file**, one of the preprocessed files is selected on the basis of detection that the selected file elaborates the **information** from the first **header file**. The first **header file** is **regenerated** from the **information** in the selected file... ... A plurality of source files and one or more **header files** are provided. The **header files** that contain **information** that several of the source files refer to. The original files are preprocessed, each to **generate** a respective preprocessed file, said preprocessing comprising **expanding** the several of the **source files** with the **information** from a first **header** 5 **file**. A collective processing step is applied to make coordinated changes to **information** from the preprocessed files. The changed preprocessed files are used to **regenerate** modified source and **header files**. For **regenerating** the first **header file**, one of the preprocessed files is selected on the basis of detection that the selected file elaborates the **information** from the first **header file**. The first **header file** is **regenerated** from the **information** in the

8/3,K/13 (Item 13 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0014280253 *Drawing available*
WPI Acc no: 2004-466764/200444
XRPX Acc No: N2004-368681
Object oriented program class implementation method in object management system, involves generating output files comprising source, interface and header files from processed information list, for each component object
Patent Assignee: EMC CORP (EMCE-N)
Inventor: BAUER A; DESAI M; MOXHAM K; PRICE T

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 6748581	B1	20040608	US 2000515773	A	20000229	200444	B

Priority Applications (no., kind, date): US 2000515773 A 20000229

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
US 6748581	B1	EN	14	6	

Alerting Abstract ...NOVELTY - An **information** list including a list of object oriented program (OOP) classes and member functions, is built. The list is processed by parsing the **header file** for each class. The output files comprising source, interface and **header files**, are automatically **generated** from the processed list, for each component object implementing member functions of a class. The output files are compiled and stored in a server.

Original Abstracts:oriented program classes as component objects in an object management system, each class including one or more member functions, each class being defined in a **header file**, builds an **information list**, the **information list** including a **list** of the classes and the member functions in each class, processes the **information** list, the

processing **including** parsing the **header file** for each **class and** reading a type file, the type file including computer code for converting between an object-oriented type and an object management type, and **generates** from the processed **information** list one or **more** output files for each component object, each component object implementing the member functions of one class.

Claims:oriented program classes as component objects in an object management system, each class including one or more member functions, each class being defined in a **header file**, the method comprising: building an **information list**, the **information** list including a list of **the** classes and **the** member functions in each class; processing the **information** list, the processing including parsing the **header file** for each **class**; automatically **generating** from the processed **information list one** or more output files for each **component** object, each component **object** implementing the member functions of one class wherein the one or more output files **include** a **source file** for each component object, an interface file for each **component** object, and **a header file** for each component object; and compiling the one or more output **files into** an object management server.

8/3,K/14 (Item 14 from file: 350)

0009591478 *Drawing available*
WPI Acc no: 1999-539744/199945
XRPX Acc No: N1999-399940
Automatic method for software build process
Patent Assignee: MCI COMMUNICATIONS CORP (MCIC-N)
Inventor: MCLAIN J V

Patent Family (1 patents, 1 countries)							
Patent Number	**Kind**	**Date**	**Application Number**	**Kind**	**Date**	**Update**	**Type**
US 5956513	A	19990921	US 1997908439	A	19970807	199945	B

Priority Applications (no., kind, date): US 1997908439 A 19970807

Patent Details					
Patent Number	**Kind**	**Lan**	**Pgs**	**Draw**	**Filing Notes**
US 5956513	A	EN	25	7	

Alerting Abstract ...NOVELTY - The program file of configuration **data** file comprising list of program **files**, **header files** and object modules is read to determine identity for each **header file**. If identified **header file** is specified in configuration **data** file, specified version is searched in specified locations. If identified **header file** is not specified most recent version of **header file** is searched. DESCRIPTION - The locations specified by configuration **data** file includes a project library and shared libraries. An internal table listing the modulus whose versions are searched in respective locations, along with module compilation **information** is **generated**. In case the identified **header file** is not found in respective locations, an unresolved conflict error message is listed in internal table. An INDEPENDENT CLAIM is also included for the automating...

...**Original Abstracts:**called an Automated Build Control (ABC) application, provides an improved tool for managing the software build process. ABC uses the specification of a programmer's **original** program **files** as **input**, and **automatically** identifies all **header**,

include, and other **shared** library **files**, which eliminates coding of dependencies. ABC identifies embedded **header files**, such as **those which** are referenced in other **header files**. ABC identifies **duplicate headers** so future builds **can** be expedited. ABC ensures all necessary source modules are located and available for compilation, prior to compilation. ABC performs comparisons of date/time stamps and... **Claims:**A method for automating a software build, said method comprising:(1) reading a configuration **data** file that provides **a** listing of modules, including program **files header files** and **object modules**, **in** order to determine which versions of said modules are required for compilation and the respective locations thereof, wherein said locations include a project library and one or more third party shared libraries;(2) scanning each said program file to determine an identity for **each header file** listed in each said **program file**;(3) if an identified **header file** is not specified in said configuration **data file**, then searching for a most recent **version** of said identified **header file** in said project library and said **third party** shared libraries specified by said configuration date file;(4) if said identified **header file** is specified in said configuration **data file**, then **searching for** said specified version of said **header file** in said project library and said third party **shared libraries** specified by said configuration date file;(5) **generating** an internal table listing said modules along with **information** relating to **the** compilation of said modules.

18/3,K/2 (Item 2 from file: 347)
DIALOG(R)File 347: JAPIO

04564407 **Image available**
DISCRIMINATION SYSTEM FOR END OF FILE
Pub. No.: 06-236307 [JP 6236307 A]
Published: August 23, 1994 (19940823)
Inventor: MIWA HISANO
Applicant: NEC SOFTWARE LTD [491061] (A Japanese Company or Corporation), JP (Japan)
Application No.: 05-019788 [JP 9319788]
Filed: February 08, 1993 (19930208)
Journal: Section: P, Section No. 1832, Vol. 18, No. 621, Pg. 132, November 25, 1994 (19941125)
International Class: G06F-012/00

ABSTRACT
PURPOSE: To recognize the end of a **file**, for which **file** end **processing** is not performed owing to system-down occurrence during the output of data to the file on a secondary storage medium, when the file is... ...CONSTITUTION: A file writing means 2 sets discrimination codes characteristic to respective **files** and **adds** the set discrimination codes to **header information** of the files and **data** records of the **files** respectively to **generate** a write **file** 31 on the secondary storage medium 3. A file reading means 4 when reading the file 31 recognizes the end of a file from the difference between the discrimination code added to the **header** of the **file** and the discrimination codes **added** to the **data** records. Di01

^ 18/3,K/12 (Item 8 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0013331152 *Drawing available*
WPI Acc no: 2003-418554/200339
XRPX Acc No: N2003-333951

Distributed file storage system using network, includes servers loaded with predefined set of software which alone enables access of stored data
Patent Assignee: TORMASOV A (TORM-I)
Inventor: TORMASOV A

Patent Family (1 patents, 1 countries)							
Patent Number	**Kind**	**Date**	**Application Number**	**Kind**	**Date**	**Update**	**Type**
US 20030041097	A1	20030227	US 2001304655	P	20010711	200339	B
			US 2002193830	A	20020711		

Priority Applications (no., kind, date): US 2001304655 P 20010711; US 2002193830 A 20020711

Patent Details						
Patent Number	**Kind**	**Lan**	**Pgs**	**Draw**	**Filing Notes**	
US 20030041097	A1	EN	16	7	Related to Provisional	US 2001304655

Original Abstracts:pieces with original file reassembly dependent only upon the number of data file pieces and not on the presence or absence of any particular data **file** piece. Local algorithms **generate** unique data **file identifiers** upon **file creation** and disassembly. Changes to the data file storage system are ranked by creation time and stored as separate records with unique transaction identifiers in addition...

..**Claims:**utilizing said local network with network servers and client computers; software constructed and arranged to run on a local client computer to enable distributed data **file** storage; software constructed and **arranged** to run on a network server computer to enable distributed **data** file storage; a **common file** namespace in the form of a tree with a shared root; directory **files** and **common data files within said common file** namespace; wherein **accessibility to** stored **data** does not depend upon dedicated access to any particular member of said **plurality** of **functionally equal network** servers, but **rather depends** only upon access to a pre-defined **quantity** of network servers from said plurality of functionally equal network servers.

18/3,K/13 (Item 9 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0013227061 *Drawing available*
WPI Acc no: 2003-311863/200330
XRPX Acc No: N2003-248311
Data structure for electronic mail file, includes separate portions for storing header, electronic mail data and variable meta-data information
Patent Assignee: DARBY D B (DARB-I); INTEL CORP (ITLC)
Inventor: DARBY D B

Patent Family (2 patents, 1 countries)							
Patent Number	**Kind**	**Date**	**Application Number**	**Kind**	**Date**	**Update**	**Type**
US 20030001889	A1	20030102	US 2001880488	A	20010613	200330	B
US 7555519	B2	20090630	US 2001880488	A	20010613	200943	E

Priority Applications (no., kind, date): US 2001880488 A 20010613

Patent Details					
Patent Number	**Kind**	**Lan**	**Pgs**	**Draw**	**Filing Notes**
US 20030001889	A1	EN	9	4	

Data structure for electronic mail file, includes separate portions for storing header, electronic mail data and variable meta-data information Alerting Abstract ...NOVELTY - **Header** containing links to **information** within the file, electronic mail **data** and variable meta-data information, are stored independently in separate portions.

Original Abstracts: An algorithm for encoded electronic mail is disclosed whereby the meta-data of the email and the files that represent the email **data** are combined into a single file. Encoding, via a **header**, is then used to monitor the changes to the meta-**data** and the location of the actual data within the **file**. The **format** of the combined **file** may be, but is not limited to, a fixed-sized **header** that contains links to the **information** within the file, followed by the actual email **data**, followed by the variable meta-**data information**. The **header** links include a link to the start of the email **data**, a link to the email sender, and a link to the email receivers. All the meta-**data information** is referenced, but not necessarily stored, in the **header**. The system is designed to maximize the use of resources including, but not limited to, memory, disk space, file descriptors, and locks... ... An algorithm for encoded electronic mail is disclosed whereby the meta-data of the email and the files that represent the email **data** are combined into a single file. Encoding, via a **header**, is then used to monitor the changes to the meta-**data** and the location of the actual data within the **file**. The **format** of the combined **file** may be, but is not limited to, a fixed-sized **header** that contains links to the **information** within the file, followed by the actual email **data**, followed by the variable meta-**data information**. The **header** links include a link to the start of the email **data**, a link to the email sender, and a link to the email receivers. All the meta-**data information** is referenced, but not necessarily stored, in the **header**. The system is designed to maximize the use of resources including, but not limited to, memory, disk space, file descriptors, and locks. **Claims:**What is claimed is:**1**. A data structure for an electronic mail file comprising:a first portion storing a **header** containing links to **information** within the file;a second portion storing electronic mail data; anda third portion storing variable meta-data information... ... into a single electronic mail file the electronic mail data and the variable meta-data information, wherein the variable meta-data describes the electronic mail **data**; monitoring changes to the variable meta-**data information** with a **header** wherein the **header** operates as an encoder and monitors changes to the variable meta-**data information**, the **header** additionally monitoring location **information** identifying the position of the electronic mail data within the electronic mail file; recompute the variable meta-data information; attempt delivery of the electronic mail file; rewrite the variable meta-**data information**; change location **information** contained within the **header**; and send the electronic mail file to a receiving computer.

18/3,K/19 (Item 15 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0012285461 *Drawing available*
WPI Acc no: 2002-226387/200228
XRPX Acc No: N2002-173736
File combination method for different computer multimedia file formats wherein the file headers are analyzed before decoding and combining the multimedia files
Patent Assignee: JUNG J (JUNG-I); KEUM G H (KEUM-I); KIM J S (KIMJ-I); KIM S H (KIMS-

I); LEE J H (LEEJ-I); WIZMAX CO LTD (WIZM-N)
Inventor: CHUNG J G; JUNG J; JUNG J G; KEUM G; KEUM G H; KIM J; KIM J S; KIM S; KIM S H; KUEM G H; LEE J; LEE J H; JUNG J K

Patent Family (6 patents, 92 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
WO 2001067277	A1	20010913	WO 2001KR27	A	20010110	200228	B
AU 200127125	A	20010917	AU 200127125	A	20010110	200228	E
KR 2001069124	A	20010723	KR 20001392	A	20000112	200228	E
KR 366760	B	20030108	KR 20001392	A	20000112	200338	E
US 20030120602	A1	20030626	WO 2001KR27	A	20010110	200343	E
			US 2002181114	A	20021112		
US 7051047	B2	20060523	WO 2001KR27	A	20010110	200635	E
			US 2002181114	A	20021112		

Priority Applications (no., kind, date): KR 20001392 A 20000112

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
WO 2001067277	A1	EN	17	8		
National Designated States,Original	AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW					
Regional Designated States,Original	AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZW					
AU 200127125	A	EN			Based on OPI patent	WO 2001067277
KR 366760	B	KO			Previously issued patent	KR 2001069124
US 20030120602	A1	EN			PCT Application	WO 2001KR27
US 7051047	B2	EN			PCT Application	WO 2001KR27
					Based on OPI patent	WO 2001067277

Alerting Abstract ...file headers of the media files to be combined are analyzed to determine the required translation algorithm. The file contents are then combined and a **file** header **generated**. For identical **file** types, the file header of the first file is modified to reflect the addition of new **data** and the **headers** of the other files are discarded. For non-identical **files**, a composite header is **generated**. USE - To combine multimedia **files** of different **formats**. ... ...ADVANTAGE - This method allows additional **information** to be **included** within the **file header** e.g. serial numbers, types of song, artist details etc. This method of combining files is also useful for Internet multimedia files as different files...

18/3,K/20 (Item 16 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0011035243 *Drawing available*
WPI Acc no: 2001-661309/200176
Method for analyzing syntax of c program
Patent Assignee: KOREA ELECTRONICS & TELECOM RES INST (KOEL-N)
Inventor: PARK S H; SHIN G S; YOON S J

Patent Family (1 patents, 1 countries)							
Patent Number	**Kind**	**Date**	**Application Number**	**Kind**	**Date**	**Update**	**Type**
KR 2001063805	A	20010709	KR 199961899	A	19991224	200176	B

Priority Applications (no., kind, date): KR 199961899 A 19991224

Patent Details					
Patent Number	**Kind**	**Lan**	**Pgs**	**Draw**	**Filing Notes**
KR 2001063805	A	KO	1	10	

Alerting Abstract ...NOVELTY - The method for analyzing the syntax of the C program is provided to exactly divide the compiling and decompiling part according to the **including** time of a **header file** by preventing the lost of the macro **information**, calibrating a parsing tree, and **separating** the user defined **header file** from a C source file. DESCRIPTION - The processing is run in order to include the macro defined sentence to the C source **file**(200). A **preprocessed file** runs parsing by using the preprocessing command grammar(210). The parsing tree is created by using pure C grammar(220). The user defined header **file** is **separated** from the parsing tree and a line number is calibrated(230).

18/3,K/31 (Item 27 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0008821530 *Drawing available*
WPI Acc no: 1998-367399/199832
Related WPI Acc No: 2004-474003
XRPX Acc No: N1998-287376
Data management apparatus for multimedia network using LAN, telephone circuit - transmits paragraph file containing paragraph information which are grouped in file divider, based on request signal
Patent Assignee: FUJITSU LTD (FUIT)
Inventor: MURAMOTO T; SAITO K; SUZUKI T; YASHIRO S

Patent Family (5 patents, 2 countries)							
Patent Number	**Kind**	**Date**	**Application Number**	**Kind**	**Date**	**Update**	**Type**
JP 10143403	A	19980529	JP 1996300705	A	19961112	199832	B
US 6370536	B1	20020409	US 1997816497	A	19970313	200227	E
US 20020049766	A1	20020425	US 1997816497	A	19970313	200233	E
US 20020083047	A1	20020627	US 1997816497	A	19970313	200245	E
			US 200282167	A	20020226		
US 6633887	B2	20031014	US 1997816497	A	19970313	200368	E

			US 200282167	A	20020226	

Priority Applications (no., kind, date): JP 1996300705 A 19961112

Patent Details						
Patent Number	**Kind**	**Lan**	**Pgs**	**Draw**	**Filing Notes**	
JP 10143403	A	JA	14	17		
US 20020083047	A1	EN			Division of application	US 1997816497
					Division of patent	US 6370536
US 6633887	B2	EN			Division of application	US 1997816497
					Division of patent	US 6370536

Alerting Abstract ...The apparatus (100) maintains a logic structure **information** file from which the **header information** is extracted. The extracted **header information** is linked with the paragraph **information** through a link file. Based on the request signal, the paragraph files containing paragraph information are grouped in a **file divider** (201) and are transmitted by a transmission controller (202...

Original Abstracts: In an information management apparatus for managing a logical structure of information file wherein a **plurality** of pieces of paragraph information, to each of which header information is appended, **are** arranged, the information management apparatus is provided with: a file dividing means for making up a **header** management **file** including **header information and** a **link in** such a manner that the **header information** is extracted **from the** logical structure of **information** file and the **link** to the paragraph **information** to which the **header information** is appended **is made** up, and also making up a plurality of paragraph **files** each **including** the **paragraph** information; **and** a transmission control means responsive to a transmission request for a logical structure of **information** file for transmitting **the header** management **file** associated **with** the **logical** structure of **information** file, and for **transmitting**, upon receipt of a transmission request based on a link **included** in the header **management file**, a paragraph **file linked** to the header management file... ... An information management apparatus for managing a logical structure of information file wherein a plurality of pieces of paragraph **information**, to each of which header information is appended, are arranged. The information management apparatus provides file dividing means for making up a header management file including header information and a link. The header information is extracted from the logical structure of **information file**, and the link to the paragraph **information to which** the **header information** is appended is **made** up. The **file dividing means** makes up a plurality of paragraph **files each including** the paragraph information. Transmission control means, **responsive** to **a** transmission request **for** a logical structure of **information** file, transmits the **header** management **file** associated with **the** logical structure of **information file and** transmits based on a link **included** in **the header** management **file**; a paragraph **file linked** to the **header** management **file**. ... ... An information management apparatus and method are provided for managing a logical structure of information file wherein a plurality of pieces of paragraph **information**, to each of which header information is appended, are arranged. The information management apparatus comprises a file dividing means for making up a header management file including header information and a link. The header information is extracted from the logical structure of **information file**, **and** the link to the paragraph **information** to **which the header information** is appended is made **up**. The

file dividing **means makes** up a plurality of paragraph **files each including** the paragraph information. A transmission control means **is** also **included** responsive to a transmission request for a logical structure **of** information file. The transmission control means transmits the header management file associated with the logical structure of **information file**, and **transmits**, upon receipt of a transmission request **based on** a link **included** in the **header** management **file**, a paragraph **file** linked **to** the header management **file**. ... ... An information management apparatus for managing a logical structure of information file wherein a plurality **of** pieces of paragraph information, to each of which header information is appended, are arranged. The information management apparatus provides file dividing means for making up a header management **file** including **header** information and a link. The **header** information is extracted **from** the **logical** structure **of information file**, and the link **to the** paragraph **information** to which the **header information is** appended is made up. The **file dividing means** makes up a plurality of paragraph **files** each **including** the paragraph **information**. **Transmission** control means, responsive to a transmission request **for** a logical structure of information file, transmits the header management file associated with the logical structure of information file and transmits based **on** a **link** included in the **header** management **file**; **a** paragraph file linked to the **header** management **file.**

18/3,K/33 (Item 29 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0007848623 *Drawing available*
WPI Acc no: 1996-478437/199648
XRPX Acc No: N1996-403512
Software computer file generation and information addition unit - includes data field with header storing identification information providing OS identity of the file
Patent Assignee: NCR CORP (NATC); NCR INT INC (NATC)
Inventor: SIEFERT D M

Patent Family (3 patents, 5 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
EP 738982	A2	19961023	EP 1996302507	A	19960410	199648	B
JP 8305618	A	19961122	JP 199698556	A	19960419	199706	E
US 5701452	A	19971223	US 1995425164	A	19950420	199806	E

Priority Applications (no., kind, date): US 1995425164 A 19950420

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
EP 738982	A2	EN	12	7	
Regional Designated States,Original	DE FR GB				
JP 8305618	A	JA	8		
US 5701452	A	EN	12	7	

Alerting Abstract ...The generation unit includes a **data** field and a **header** in which identification **information** is stored, other than the **information** required in the presentation of any data stored in the data field on a computer display screen... ...natural language of the file, a type encryption of the file, a type compression of the file. This provides an identity of the program which **generated** the **file**. Providing identity of the

operating system upon which the generation program ran. Communication protocol information is also provided...

Original Abstracts: The invention relates to means for **generating** a computer file **comprising** a **data field** and a **header**. The invention is **characterized** in that in the **header** is stored identification **information** other than the **information** required in the **presentation** of any **data** stored in the data field on a computer display screen... ... The invention concerns a file structure for a computer file. A computer file normally contains (a) a system **header**, (b) a **header generated** by an application program, **and** (c) **information generated** by a user. The invention creates **information in** addition to these three types, and **adds** it to the **file**. This **added** information is **used** by other programs, **to** transform **the** data, prior to presentation to the user. **Claims:**1. Means for **generating** a **computer file**, characterized **in** that said file comprises a **data** field **and** a **header** in **which** is stored identification **information** other **than** the **information** required **in** the presentation of any data stored in said data field on a computer display screen... ... Claim 6. A method of **processing** a computer **file** having **file** data, comprising **the** following steps:**a**) maintaining, **in** association with the file, information indicative of at least two of the following:i) natural language of the file;ii) type of encryption of the file;iii) type of compression of the file;iv) identity of the program which **generated** the **file**, which program runs on an operating system;v) identity of the operating system upon which **the generating program** ran; andvi) communication protocol required by the file;b) locating the information indicative of at least two of i) through vi) between a header for the system and a **file header** for the file **data**; **and**c) reading said **information** and calling respective programs which **process** the **file** in accordance with the **information**, **without human** intervention.

18/3,K/34 (Item 30 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0007713281 *Drawing available*
WPI Acc no: 1996-336069/199634
XRPX Acc No: N1996-283256
Host to portable terminal data communications for insurance sales - involves setting data record structures in which communications-specified fields are embedded to allow host to carry out batch or real-time data processing
Patent Assignee: TREND INVESTMENTS LTD (TREN-N)
Inventor: MURRAY C; PEELO F

Patent Family (2 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
GB 2297400	A	19960731	GB 19951654	A	19950127	199634	B
GB 2297400	B	19990804	GB 19951654	A	19950127	199933	E

Priority Applications (no., kind, date): GB 19951654 A 19950127

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
GB 2297400	A	EN	23		

Alerting Abstract ...The method involves a communications controller transmitting signals to a host, setting a data record structure in which communication specific fields are embedded. The host **generates** an output **file** which may be used for further communications. The controller receives a **common-format file** with a preset structure **including data** records from the host. Records are sorted into groups and written to memory... **Class Codes** International Patent Classification IPC Class Level Scope Position Status Version Date **G06F-0013/38**... **G06F-0013/38**...

Claims: The method involves a communications controller transmitting signals to a host, setting a data record structure in which communication specific fields are embedded. The host **generates** an output **file** which may be used for further communications. The controller receives a **common-format file** with a preset structure **including data** records from the host. Records are sorted into groups and written to memory...

~ ~ Patent Literature Full-Text

DIALOG(R)File 348: EUROPEAN PATENTS

9/3K/1 (Item 1 from file: 348)
02897267

METHOD AND STRUCTURE TO DEVELOP A TEST PROGRAM FOR SEMICONDUCTOR INTEGRATED CIRCUITS

VERFAHREN UND STRUKTUR ZUR ENTWICKLUNG EINES TESTPROGRAMMS FUR INTEGRIERTE HALBLEITERSCHALTUNGEN

PROCEDE ET STRUCTURE POUR ELABORER UN PROGRAMME D'ESSAI POUR CIRCUITS INTEGRES A SEMI-CONDUCTEUR

Patent Assignee:

- **ADVANTEST CORPORATION** (520936)
  1-32-1, Asahi-cho; Nerima-ku,Tokyo 179-0071 (JP)
  (Proprietor designated states: all)

Inventor:

- **PRAMANICK, Ankan**
  5839 Capilano Drive,; San Jose, California 95318; (US)
- **ELSTON, Mark**
  1777 Merlot Way; Salinas, California 93906; (US)
- **KRISHNASWAMY, Ramachandran**
  21327 Meteor Drive; Cupertino, California 95014; (US)
- **ADACHI, Toshiaki**
  1196 Nikette Way; San Jose, California 95120; (US)

Legal Representative:

- **Nicholls, Michael John (61941)**
  J.A. Kemp & Co. 14 South Square; Gray's InnLondonWC1R 5JJ; (GB)

	Country	Number	Kind	Date	
Patent	EP	1756601	B1	20091209	(Basic)
	WO	2005114235		20051201	
Application	EP	2005743229		20050523	
	WO	2005JP9813		20050523	
Priorities	US	573577	P	20040522	
	US	918714		20040813	

Specification: ...directing each module-specific compiler to compile both a corresponding module-specific section of a pattern source file and a common section of the pattern **source file**. The **common** section **includes information** accessible to all of the module-specific compilers. An output of the compiler includes at least one module-specific pattern data

section. Module-specific pattern... ...system further includes means for compiling the test plan file and the pre-header file to form a derived test plan file and a header **file** respectively, and means for **generating** a test program using the test class file, the derived test plan file, and the header file.

9/3K/3 (Item 3 from file: 348)
00902640

Locking system for coupling modules to a modular computer
Verriegelungssystem zur Verbindung von Modulen zu einem modularen Rechnersystem
Systeme de verrouillage pour coupler des modules a un ordinateur modulaire

Patent Assignee:

- **TEXAS INSTRUMENTS INCORPORATED** (279070)
  13500 North Central Expressway; Dallas Texas 75265 (US)
  (Proprietor designated states: all)

Inventor:

- **Leavitt, Thomas**
  3215 Poplar Road; Temple, TX 76502; (US)
- **Harvey, David L.**
  709 North 5th Street; Temple, TX 76501; (US)
- **Shin, Seong**
  11656 Cedar Spring Court; St. Cupertino, CA 95014; (US)

Legal Representative:

- **Legg, Cyrus James Grahame et al (81121)**
  ABEL & IMRAY, 20 Red Lion Street; London WC1R 4PQ; (GB)

	Country	Number	Kind	Date	
Patent	EP	823686	A2	19980211	(Basic)
	EP	823686	A3	19990407	
	EP	823686	B1	20030219	
Application	EP	97304113		19970612	
Priorities	US	663499		19960614	

Specification: ...changed to operate under different operating systems. Name of Program : purpose AboutDlg.h : header file AboutDlg.cpp : implementation file MainWnd.cpp : implementation file MainWnd.h : **header file** MBMOn.clw : CLW file contains **information** for MFC ClassWizard MBMon.cpp : Defines the class behaviors for the application. MBMon.h : main header file for the MBMON application MBMon.rc : Microsoft(TM... ...TM) Visual C++ RemoveMedia.cpp : implementation file RemoveMedia.h : header file RemovingDriver.cpp : implementation file RemovingDriver.h : header file resource.h : Microsoft(TM) Developer Studio **generated** include **file**. SMIDlg.cpp : implementation file SMIDlg.h : header file

stdafx.cpp : **source file** that **includes** just the standard includes stdafx.h : include file for standard system include files, stdtypes.h : from Windows.h TISYS.H : header file for interface to...

Specification: ...changed to operate under different operating systems. Name of Program : purpose AboutDlg.h : header file AboutDlg.cpp : implementation file MainWnd.cpp : implementation file MainWnd.h : **header file** MBMon.clw : CLW file contains **information** for MFC ClassWizard MBMon.cpp : Defines the class behaviors for the application. MBMon.h : main header file for the MBMON application MBMon.rc : Microsoft(TM... ...TM) Visual C++ RemoveMedia.cpp : implementation file RemoveMedia.h : header file RemovingDriver.cpp : implementation file RemovingDriver.h : header file resource.h : Microsoft(TM) Developer Studio **generated** include **file**. SMIDlg.cpp : implementation file SMIDlg.h : header file stdafx.cpp : **source file** that **includes** just the standard includes stdafx.h : include file for standard system include files, stdtypes.h : from Windows.h TISYS.H : header file for interface to...

9/3K/4 (Item 4 from file: 348)
00901346

System for managing messages
System fur Nachrichtverwaltung
Systeme pour la gestion de messages

Patent Assignee:

- **SUN MICROSYSTEMS, INC.** (1392732)
  2550 Garcia Avenue; Mountain View, California 94043-1100 (US)
  (applicant designated states:
  AT;BE;CH;DE;DK;ES;FI;FR;GB;GR;IE;IT;LI;LU;MC;NL;PT;SE)

Inventor:

- **Walls, Douglas**
  1009 Lassen Drive; Belmont, CA 94002; (US)
- **Walker, Kathryn**
  10981 Northsky Square; Cupertino, CA 95014; (US)
- **Mori, Robert F.**
  937 Middle Avenue; Menlo Park, CA 94025; (US)

Legal Representative:

- **Hogg, Jeffery Keith et al (31905)**
  Withers & Rogers 4 Dyer's Buildings Holborn; London EC1N 2JT; (GB)

	Country	Number	Kind	Date	
Patent	EP	822483	A2	19980204	(Basic)
Application	EP	97303956		19970609	

	Country	Number	Kind	Date
Priorities	US	680499		19960701

Specification: ...could not handle the underscore character and is case insensitive. The symbolic handle 62 is therefore converted into a help system topic ID 64. The **header files** contain all the **information** needed by the compiler for any messages.

The definition of the compiler interface routines used in the preferred embodiment to access **information** on a message stored in the **header files** 12 produced from the master message file 10 are set out in Fig. 8. These interface routines are defined in a single message **source file** which also **includes** the message array 12(2). Each of these interface routines return information from an array element indexed by the message number (msgno) provided as an... ...compiler to generate error messages and links to online help messages. If an error occurs when compiling a program the compiler utilizes the enumeration header **file** 12(1) to **generate** an enumerated symbolic handle indexing the array element including the error message. The enumerated symbolic handle is utilized to access an array element including all...

Claims: ...message text, and help message text, the message management method comprising the steps of:

generating a help volume, based on information in said master message **file**, utilized to generate online help;

generating **header files**, based on **information** in said master message file, **included** in a **source file** utilized to generate messages; and

generating a message catalog **file**, based on said information in said master message file, utilized to localize the message text.

2. A computer implemented message management method based on a... ...text, the message management method comprising the steps of:

producing compiler header files from said master message file;

providing a symbolic handle of required message **information** to be displayed;

processing said provided symbolic handle and said **header files** to access said required **information**;

displaying the required message **information** accessed from said **header files**;

producing a help volume from said master message file, with said help volume having a plurality of entries each including help message text and a...option text, and option help text, the option management method comprising the steps of:

generating a help volume, based on information in said master option **file**, utilized to generate online help;

generating header files, based on **information** in said master option file, **included** in a **source file** utilized to **generate** option text; and

generating an option catalog **file**, based on said information in said master option file, utilized to localize the option text.

9/3K/8 (Item 4 from file: 349)
DIALOG(R)File 349: PCT FULLTEXT

01529540

METHOD AND SYSTEM FOR SORTING/ SEARCHING FILE AND RECORD MEDIA THEREFOR
PROCEDE ET SYSTEME DESTINES A TRIER/RECHERCHER DES FICHIERS ET SUPPORT D'ENREGISTREMENT ASSOCIE

Patent Applicant/ Patent Assignee:

- **NHN CORPORATION**
  Bundang Venture Town, 25-1, Jeongja-dong, Bundang-gu, Seongnam-si, Kyunggi-do 463-844; KR; KR (Residence); KR (Nationality); (For all designated states except: US)

Patent Applicant/ Inventor:

- **CHANG Joon-Kee**
  #102-302 Hyundaifeelgreen, Gyodongmaeul, Mabuk-dong, Giheung-gu, Youngin-si, Gyeonggi-do 449-561; KR; KR (Residence); KR (Nationality)
- **SO Soon-Sik**
  #501, Indang Bldg., 163-4 Songpa 2-dong, Songpa-gu, Seoul 138-172; KR; KR (Residence); KR (Nationality)
- **KIM Mi-Yeon**
  #101, Hoyeartvilla, 646-40 Jayang 2-dong, Gwangjin-gu, Seoul 143-192; KR; KR (Residence); KR (Nationality)

Legal Representative:

- **LEE Kyeong-Ran (agent)**
  502, BYC Bldg., 648-1 Yeoksam 1-dong, Kangnam-ku, Seoul 135-081; KR

	Country	Number	Kind	Date
Patent	WO	200773097	A1	20070628
Application	WO	2006KR5606		20061220
Priorities	KR	1020050126407		20051220

Detailed Description:

...skilled in the art that one integrated application can perform the above-described processes.

In accordance with another embodiment of the present invention, an annotation **file** can be **generated** for a **file** that is created by a user on a web and uploaded to the web server 102. The web server 102 can receive a search keyword from the user and search the original file or the annotation file.

In case the user selects a created **file** and request the **generation** of an annotation **file** on the web, the web server 102 in accordance with a second embodiment of the present invention transmits a webpage for inputting annotation details and carries out the steps of **generating** an annotation **file** after receiving the annotation details inputted by the user.

Then, in case that the web server 102 receives a search keyword from the user, the... ...the first embodiment of the present invention.

As illustrated in FIG. 2, the annotation application in accordance with the present invention can include an annotation **file generating** module 200, an annotation **file** editing module 202, a file outputting module 204 and a user interface module 206.

The annotation **file generating** module 200 **generates** an annotation **file** in accordance with annotation details inputted by the user.

In case that the user inputs the annotation details into the annotation interface, the annotation **file generating** module 200 **generates** an annotation **file** having the annotation details. The annotation file in accordance with the present invention is defined with an extensible markup language (XML) as illustrated in FIG... ...le> bright and clear day Nickon

15/3K/6 (Item 6 from file: 348)
01072790

System and method for reducing the footprint of preloaded classes
System und Verfahren zum reduzieren des Platzbedarfs von vorgeladenen Klassen
Systeme et procede pour reduire la taille des classes prechargees

Patent Assignee:

- **SUN MICROSYSTEMS, INC.** (1392733)
  901 San Antonio Road Palo Alto,; California 94303 (US)
  (Applicant designated States: all)

Inventor:

- **Kawahara, Hideya**
  520 Walker Drive; Mountain View, California 94043; (US)
- **Fresko, Nedim**
  1366 5th Avenue; San Francisco,California 94122; (US)

Legal Representative:

- **Cross, Rupert Edward Blount et al (42891)**
  BOULT WADE TENNANT, Verulam Gardens 70 Gray's Inn Road; London WC1X 8BT; (GB)

	Country	Number	Kind	Date	
Patent	EP	943989	A2	19990922	(Basic)
	EP	943989	A3	20050928	
Application	EP	99302198		19990322	
Priorities	US	45508		19980320	

Specification: ...the system of the present invention include a collection of class files, a Java class preloader in which the above method is implemented and output **files generated** by the preloader, **including** preloaded classes, header **files** and **source** code **files**.

The class files define the complete set of classes to be preloaded. The preloader performs a first pass on the class files to determine the... ...files are compatible with similar files employed by conventional Java systems. That is, the pre-loaded classes can be assembled or compiled into class object **data** and the **header files** and source files can be compiled with VM sources into VM object data. The VM and class object data can then be linked in the... ...more readily apparent from the following detailed description and appended claims when taken in conjunction with the drawings, in which:

~~ Non-Patent Literature Abstracts

7/3,K/1 (Item 1 from file: 35)
DIALOG(R)File 35: Dissertation Abs Online

01466139 ORDER NO: AADAA-IMM01959
VISUALIZING THE SOURCE FILE STRUCTURE OF SOFTWARE WRITTEN IN C
Author: TZERPOS, VASSILIOS
Degree: M.SC.
Year: 1995
Corporate Source/ Institution: UNIVERSITY OF TORONTO (CANADA) (0779)
Source: Volume 34/02 of MASTERS ABSTRACTS. of Dissertations Abstracts International. PAGE 796 . 96 PAGES
ISBN: 0-315-01959-4

...systems is a technique that can help developers understand and improve their systems.

We present SoFi (for Source File), a tool that extracts source file **information** and presents it in a visual manner. We show that for large systems visualization should be accompanied by clustering facilities in order to manage complexity. SoFi provides such a facility that produces diagrams where nodes may also represent subsystems (collections of **files**).

A **common** anomaly in C systems is the unnecessary inclusion of source code. We present a method of discovering such gratuitous dependencies along with other poor design styles such as "sneak" references (with no explicit "**includes**") between **source files**. We propose a well-formedness theory for C "includes" and examine whether developers of various systems follow the rules of this theory.

The results of...

12/3,K/2 (Item 2 from file: 2)
DIALOG(R)File 2: INSPEC

09189773
Title: RTL/ ISS co-modeling methodology for embedded processor using SystemC
Author(s): Yuyama, Y.; Aramoto, M.; Kobayashi, K.; Onodera, H.
Author Affiliation: Dept. of Commun. & Comput. Eng., Kyoto Univ., Japan
Book Title: 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)
Inclusive Page Numbers: V-305-8 Vol.5
Publisher: IEEE, Piscataway, NJ
Country of Publication: USA
Publication Date: 2004
Conference Title: 2004 IEEE International Symposium on Circuits and Systems
Conference Date: 23-26 May 2004
Conference Location: Vancouver, BC, Canada
Conference Sponsor: IEEE Circuits and Syst. Soc
ISBN: 0 7803 8251 X
U.S. Copyright Clearance Center Code: 0 7803 8251 X/2004/$17.00
Part: Vol.5
Number of Pages: 5537
Language: English

Subfile(s): B (Electrical & Electronic Engineering); C (Computing & Control Engineering)
INSPEC Update Issue: 2004-046

Title: RTL/ ISS co-modeling methodology for embedded processor using SystemC
Abstract: We propose ISS/RTL co-modeling methodology by describing both in **common source file** using SystemC. Our method enables rapid and easy **generation**/verification of RTL/ISS of customizable **processor**. We apply this method to **processor** "MiU-**Processor**". As a result coded-sharing ratio is 67%. For adding new instruction, we add only 12 lines to RTL/ISS shared apart. Our ISS **generation** method is very effective for multi customizable SoC
Identifiers: RTL/ISS co-modeling technology; embedded **processor**; SystemC; **common source file**; MiU-**Processor**; coded-sharing ratio; ISS **generation** method; SoC; instruction set simulator; multicustomizable **processor**

12/3,K/3 (Item 3 from file: 2)
DIALOG(R)File 2: INSPEC

09128443
Title: Distributed loadbuilds
Author(s): Smith, K.W.
Journal: Dr. Dobb's Journal , vol.28 , no.7 , pp.58-60
Publisher: CMP Media LLC
Country of Publication: USA
Publication Date: July 2003
ISSN: 1044-789X
SICI: 1044-789X(200307)28:7L.58:DL;1-J
CODEN: DDJSDM
Language: English
Subfile(s): C (Computing & Control Engineering)
INSPEC Update Issue: 2004-040

Abstract: The idea of **dividing** a computationally intensive task between multiple physical machines has been around for as long as computer networks have existed. There are a variety of toolkits... ...are not impacted by a code change are not recompiled. Unfortunately, there are situations where build avoidance is not enough. For example, changing low-level **header files** might force recompiles of all **source files** in the system. Also, there are times when you want to build cleanly to ensure you have a stable configuration. These situations are ideal candidates...
Descriptors: distributed **processing**; resource allocation; Unix

12/3,K/6 (Item 6 from file: 2)
DIALOG(R)File 2: INSPEC

05436892
Title: Reducing compilation time by a compilation server
Author(s): Onodera, T.
Author Affiliation: Tokyo Res. Lab., IBM Res., Japan
Journal: Software - Practice and Experience , vol.23 , no.5 , pp.477-85
Country of Publication: UK

10552777 Processing of a Compileable Computer Program - Results

Publication Date: May 1993
ISSN: 0038-0644
CODEN: SPEXBL
U.S. Copyright Clearance Center Code: 0038-0644/93/050477-09$09.50
Language: English
Subfile(s): C (Computing & Control Engineering)
INSPEC Update Issue: 1993-025

Abstract: In language systems that support **separate** compilation, one often observes that **header files** are internalized over and over again when the **source files** that depend on them are compiled. Making a compiler a long-lived server eliminates such redundant **processing** of **header files**, thus reducing the compilation time. The paper first describes compilation servers for C-family languages in general, and then a compilation server for a C...
Identifiers: compilation time; compilation server; language systems; **separate** compilation; **header files**; long-lived server; C-family languages; C-based object-oriented language

12/3,K/10 (Item 1 from file: 35)
DIALOG(R)File 35: Dissertation Abs Online

01650632 ORDER NO: AAD98-35636
VIRTUAL ENTITIES AS A MODEL FOR THE TRANSMISSION OF MULTIMEDIA ACROSS NETWORKS
Author: ELMAHDY, HESHAM NABIH
Degree: PH.D.
Year: 1997
Corporate Source/ Institution: THE UNIVERSITY OF MISSISSIPPI (0131)
Source: Volume 5906B of Dissertations Abstracts International.
PAGE 2846 . 123 PAGES

The design of a multimedia file **format** that takes into consideration improving the performance of transmission of multimedia across networks is addressed. This dissertation presents the concept of a Virtual Entity for Multimedia (VEM). The VEM model was designed to accommodate all raster data graphic file **formats**. The VEM model also **separates** the **header** of the **source** graphic **file** from the source data creating two or more files. One file contains the source description data in our VEM **header format**. The VEM **header** contains the name of data file(s) and offsets to extension blocks. Source raster data is decoded (if it was encoded). Next, an appropriate archiving algorithm is selected and applied to the source data and the archive algorithm name is saved in the VEM **header**. After applying this algorithm, the data is saved in a second (data) file.

By using VEM, the number of conversion programs required among "N" multimedia file **formats** are reduced from N*(N $-$ 1) into 2*N. VEM improves the performance of transmission of multimedia files across networks by reducing the size of... ...on multimedia files, it eliminates the possibility of corrupting multimedia files due to encoding/decoding compatibility problems, and it permits addition of new multimedia file **formats** without recompiling the destination converting program.

12/3,K/13 (Item 1 from file: 99)
DIALOG(R)File 99: Wilson Appl. Sci & Tech Abs

10552777 Processing of a Compileable Computer Program - Results

1286135 **H.W. Wilson Record Number:** BAST96008309
Create your own compile time diagnostics
Richards, Robert ;
Electronic Design v. 44 (Jan. 22 '96) p. 126
Document Type: Feature Article **ISSN:** 0013-4872

Abstract: Part of a special section on engineering software for software design engineers. The writer describes how to create compile time diagnostics. The **process** of optimizing code in an embedded controller can result in an error-prone environment when tasks are **split** across several **source files**. Ensuring that any changes made to one file do not invalidate any assumptions made in other **source files** can usually be accomplished using compile-time constants in a **common header file**. The compiler can be used to verify any assumptions so that many types of bugs can be eliminated at compile time. It is important that...

16/3,K/1 (Item 1 from file: 2)
DIALOG(R)File 2: INSPEC

09543658
Title: C++/CLI: headers, inline functions, arrays and generics
Author(s): Jaeschke, R.
Journal: C/C++ Users Journal , vol.23 , no.2 , pp.36-41
Publisher: CMP Media Inc
Country of Publication: USA
Publication Date: Feb. 2005
ISSN: 1075-2838
SICI: 1075-2838(200502)23:2L.36:HIFA;1-Z
CODEN: CCUJEX
Language: English
Subfile(s): C (Computing & Control Engineering)
INSPEC Update Issue: 2005-034

Title: C++/CLI: headers, inline functions, arrays and generics
Abstract: In traditional C++ design and implementation, we define each type being modeled, in its own **header**, with that **header** containing the type's name, its members' names and types, and the inline definition of relatively trivial member functions. Rather than having **separately** compiled **source files** share **information** via a **header**, in C++/CLI such **information** is shared via an assembly. As with the Point class, it was compiled on its own, resulting in an assembly called "Point.dll". The author...
Descriptors: C++ language; **data** structures; object-oriented programming
Identifiers: C++ language; CLI **headers**; inline function; arrays; Point class extensions; **data** structures; object-oriented programming

16/3,K/5 (Item 1 from file: 8)
DIALOG(R)File 8: Ei Compendex(R)

1210245461 **E.I. COMPENDEX No:** 20093412267324
Dependability improvement for PPM compressed data by using compression pattern matching
Kitakami, Masato; Okura, Toshihiro

Corresp. Author/ Affil: Kitakami, M.: Graduate School of Advanced Integration Science, Chiba University, Chiba-shi, 263-8522, Japan
Corresp. Author email: kitakami@faculty.chiba-u.jp
IEICE Transactions on Information and Systems (IEICE Trans Inf Syst) (Japan) 2008 E91-D/10 (2435-2439)
Publication Date: 20081001
Publisher: Maruzen Co., Ltd.
CODEN: ITISE **ISSN:** 0916-8532 **eISSN:** 1745-1361
Item Identifier (DOI): 10.1093/ietisy/e91-d.10.2435
URL: http://ietisy.oxfordjournals.org/cgi/reprint/E91-D/10/2435
Document Type: Article; Journal **Record Type:** Abstract
Language: English **Summary Language:** English
Number of References: 10

Dependability improvement for PPM compressed data by using compression pattern matching

Data compression is popularly applied to computer systems and communication systems in order to reduce storage size and communication time, respectively. Since large **data** are used frequently, string matching for such **data** takes a long time. If the **data** are compressed, the time gets much longer because decompression is necessary. Long string matching time makes computer virus scan time longer and gives serious influence to the security of **data**. From this, CPM (Compression Pattern Matching) methods for several compression methods have been proposed. This paper proposes CPM method for PPM which achieves fast virus scan and improves dependability of the compressed **data**, where PPM is based on a Markov model, uses a context **information**, and achieves a better compression ratio than BW transform and Ziv-Lempel coding. The proposed method encodes the context **information**, which is **generated** in the compression process, and appends the encoded **data** at the beginning of the compressed **data** as a **header**. The proposed method uses only the **header information**. Computer simulation says that augmentation of the compression ratio is less than 5 percent if the order of the PPM is less than 5 and the **source file** size is more than 1 M bytes, where order is the maximum length of the context used in PPM compression. String matching time is independent of the **source file** size and is very short, less than 0.3 micro seconds in the PC used for the simulation. Copyright (c) 2008 The Institute of Electronics, **Information** and Communication Engineers.
Descriptors: Communication systems; Compression ratio (machinery); Computational methods ; Computer simulation languages; Computer software; Computer viruses; **Data** compression ratio; Markov processes; Pattern matching; Phase modulation; Pulse modulation; Pulse position modulation; Security of **data**; * **Data** compression

16/3,K/6 (Item 2 from file: 8)
DIALOG(R)File 8: Ei Compendex(R)

0013021915 **E.I. COMPENDEX No:** 1993081013254
Reducing compilation time by a compilation server
Onodera, Tamiya
Corresp. Author/ Affil: Onodera, Tamiya: Tokyo Research Lab, Tokyo, Japan
Software - Practice and Experience (Software Pract Exper) 1993 23/5 (477-485)
Publication Date: 19930101
CODEN: SPEXB **ISSN:** 0038-0644
Document Type: Article; Journal **Record Type:** Abstract
Treatment: A; (Applications); G; (General review); T; (Theoretical)
Language: English **Summary Language:** English

Number of References: 9

In language systems that support **separate** compilation, we often observe that **header files** are internalized over and over again when the **source files** that depend on them are compiled. Making a compiler a long-lived server eliminates such redundant processing of **header files**, thus reducing the compilation time. The paper first describes compilation servers for C-family languages in general, and then a compilation server for our C...

Descriptors: C (programming language); Computer programming; **Data** processing; File organization; Program compilers; Program processors; *Computer systems

Identifiers: C based objects; Compilation servers; **Header files**

16/3,K/8 (Item 2 from file: 60)
DIALOG(R)File 60: ANTE: Abstracts in New Tech & Engineer

0001603894 IP Accession No: 20080922094
System and method for retrieving registry data
Gray, David Michael; Glerum, Kirk A; Ratan, Maithreyi Lakshmi
, USA
Publisher Url: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=/netaht ml/PTO/search-adv.htm&r=1&p=1&f=G&l=50&d=PTXT&S1=7379948.PN.&OS=pn/7379948& RS=PN/7379948
Document Type: Patent
Record Type: Abstract
Language: English
File Segment: ANTE: Abstracts in New Technologies and Engineering
System and method for retrieving registry data
Abstract:
Information is retrieved from the registry by creating a registry interface database containing **information** regarding registry **data** and uniquely identifying every registry entry for a registry interface. **Source header files** are **generated** using **information** about a program module, such as an application, contained in the registry interface database. The **source header files** are compiled with the application to **generate** the application's executable file. During run-time of the application, the registry interface is called by the application to retrieve registry **data**. If available, the registry interface retrieves the registry **data** and forwards this **information** to the application. In the event that the registry **data** is not available, the registry interface retrieves a default value in place of the registry **data** and forwards that default **data** to the application. A policy may be implemented by checking a policy area of the registry for the registry **data** and invoking the policy.
Descriptors: Databases; **Header files**; Run time (computers); Modules
Identifiers:

16/3,K/10 (Item 4 from file: 60)
DIALOG(R)File 60: ANTE: Abstracts in New Tech & Engineer

0000671233 IP Accession No: 2008384733
Embedded flow information for binary manipulation
Richardson, Stephen E
, USA
Publisher Url: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=/netaht ml/PTO/search-adv.htm&r=1&p=1&f=G&l=50&d=PTXT&S1=59

26639.PN.&OS=pn/5926639& RS=PN/5926639
Document Type: Patent
Record Type: Abstract
Language: English
File Segment: ANTE: Abstracts in New Technologies and Engineering
Embedded flow information for binary manipulation

Abstract:
A method and apparatus for making flow **information** available for binary manipulation tasks are disclosed. Flow **information** is **generated** and saved either by a compiler or by a flow **information generator**. A compiler **generates** the flow **information** directly from a **source file** while the compiler is compiling the **source file** into an executable file. A flow **information generator generates** the flow **information** from an executable file in a manner similar to a compiler. Further, the flow **information generator** groups the executable file into units of text and traces the units to produce the flow **information**. The binary **information** thus retrieved is saved and embedded either in a text or a **header** of the executable file or placed in a file **separate** from the executable file. The flow **information** may be used in binary manipulations including binary translations, binary-to-binary optimizations, program tracing, and program debugging.
Descriptors: Compilers; **Generators**; Texts; Mountains; Sun; Translations; Debugging; **Headers**

~~ Non-Patent Literature Full-Text

8/3,K/3 (Item 2 from file: 47)
DIALOG(R)File 47: Gale Group Magazine DB(TM)

03694651 **Supplier Number:** 11712724 (USE FORMAT 7 OR 9 FOR FULL TEXT)
StratosWare Corp.: MemCheck 2.1. (Software Review) (one of two evaluations of debugging/ testing software in 'Debugging Tools Tame Overwrite Bugs; C-Debug, MemCheck Insert Debugging Statements Directly into a Program's Source Code') (Evaluation)

Schulz, Peter
PC Week , v9 , n2 , p92(2)
Jan 13 , 1992
Document Type: Evaluation
ISSN: 0740-1604
Language: ENGLISH **Record Type:** FULLTEXT; ABSTRACT
Word Count: 829 **Line Count:** 00066
...to the source code must be made in order for MemCheck to function with the program that the developer is trying to debug. The MemCheck **header file** must be included in every source module, and calls to MemCheck's start and end routines must be inserted manually as the first and last lines of code to be executed by the program. The **header-file** insertion is h andled automatically by the MCCONFIG program. The file or files that are to be converted can be specified by name, by wild-card reference or by a file containing a list of the files to be converted. MCCONFIG does expect all the Include **files** (or **header files**) in the source code to be located at the beginning, before any local **data** definitions or executable code. This restriction caused MCCONFIG to give a warning about a complicated **source** code **file** that **included** code fragments. However, ignoring these warni ngs during testing caused no problems. The developer merely has to make a simple manual modification to add the...

8/3,K/10 (Item 4 from file: 275)
DIALOG(R)File 275: Gale Group Computer DB(TM)

01429687 **Supplier Number:** 10644454 **(Use Format 7 Or 9 For** FULL TEXT)
Borland C++ 2.0 provides a comprehensive Windows development environment. (Borland International Inc.'s compiler, Microsoft Corp.'s graphical user interface) (Software Review) (evaluation)
Shaw, Richard Hale
Microsoft Systems Journal , v6 , n3 , p45(12)
May , 1991
Document Type: evaluation
ISSN: 0889-9932
Language: ENGLISH **Record Type:** FULLTEXT; ABSTRACT
Word Count: 7247 **Line Count:** 00589
...symbol table. The next time the source file is compiled, the symbol table is loaded directly off the disk, rather than having to recompile the **headers**. The scenario is the same for other source files that use

the **header files**. Even if the original source is changed, the precompiled **headers** are available as long as the file references one or more of the same **header files**. Borland claims that reading symbol table files is ten times faster than recompiling the **headers**.

There are several caveats, of course. First, the **headers** themselves must not have changed since the symbol table was first compiled. The time/date stamp of each **header** is checked against the **information** in the symbol table. And before the first # **include** in the **source file**, the same macros must be defined to the same values. Third, since **header** order can affect the content of macro definitions, the **headers** must occur in the same order in the source file. Finally, the compiler options and settings need to be identical for each compilation; specifically, the...the second and subsequent compiles (I ran five of each), BCC repeatedly took less than 2.5 seconds, and BCCX only slightly more. With precompiled **headers**, both Borland compilers took less than half the time required by a Microsoft C compile with optimization turned off.

When you're compiling a C...

...compilers delivered compile times between 14 to 16 seconds to compile an application for Windows written in C++. When they built the symchecked against the **information** in the symbol table. And before the first # **include** in the **source file**, the same macros must be defined to the same values. Third, since **header** order can affect the content of macro definitions, the **headers** must occur in the same order in the source file. Finally, the compiler options and settings need to be identical for each compilation; specifically, the...